基于意图的物联网服务描述与发现 *

刘星宇 [a]，江凌云 [a, b†]

(南京邮电大学 a. 通信与信息工程学院; b. 物联网研究院，南京 210003)

摘 要：在物联网服务发现过程中，用户通常用自己的意图表达需求，而服务描述是对服务功能的说明，因此两者之间的不匹配会影响服务发现的准确率。同时随着服务的种类不断增多，服务发现的准确率不断降低。为了解决上述问题，文章提出在物联网服务描述中引入意图服务本体的方法，并在意图服务本体中扩展服务上下文和Qos(Quality of Service)。扩展后的意图服务本体存储在语义化服务描述 OWL-S(Ontology Web Language for Services)文件中，能够以意图的方式表达服务功能，丰富物联网服务描述的语义，提高服务发现的准确率。仿真结果表明，文章所提出的服务描述方法以及相应的服务发现算法相对于传统服务发现方法能提升 6.7%的准确率。

关键词：意图服务本体；上下文；Qos(Quality of Service)；服务描述；服务发现

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2022.02.0044

Intent-based IoT service description and discovery

Liu Xingyu[a], Jiang Lingyun[a, b†]

(*a. School of communication & information engineering*, *b. Institute of Internet of Things*, *Nanjing University of Posts & Telecommunications*, *Nanjing* 210003, *China*)

Abstract: In the process of Internet of Things service discovery, users usually express their needs with their own intentions, while service description is the description of service functions, so the mismatch between the two will affect the accuracy of service discovery. At the same time, the accuracy of service discovery decreases with the increasing of service types. In order to solve the above problems, this paper proposed a method of introducing intentional service ontology into the description of iot services, and extended the service context and Qos (Quality of Service) in the intentional service ontology. The extended intentional service ontology was stored in OWL-S(Ontology Web Language for Services) files, which can express service functions in an intentional way, enrich the semantics of iot service description, and improve the accuracy of service discovery. Simulation results showed that the proposed service description method and the corresponding service discovery algorithm can improve the accuracy of 6.7% compared with the traditional service discovery method.

Key words: intended service ontology; context; QoS(quality of service); service description; service discovery

0 引言

现如今物联网技术已成为社会各个产业和行业中不可或缺的一部分，物联网是在互联网的基础上引入“物”的概念，将互联网与人们生活中的各种物品、设备联系在一起。随着物联网技术的不断发展，物联网服务种类不断增多，服务发现准确率相应不断降低，通用的服务描述和发现机制不能满足人们的需求，因此本文提出改进的服务描述与服务发现算法。

本体是语义网的基础，是包含了某个学科领域内的概念以及概念间的关系的模型。在服务描述语言中，由于WSDL(Web Services Description Language)[1]基于关键字的服务匹配方式准确率不高，研究者们提出语义化服务描述语言。现有的语义服务描述语言包括 OWL-S(Web Ontology Language for Service)[2]、WSMO((Web Semite Frameworking Ontology)[3]和 WSDL-S[4]等。OWL-S 通过引入本体的概念实现服务描述从语法级(WSDL)到语义级(OWL-S)的转换，是一种标准的语义标记语言，且 OWL-S 作为机器能够理解的语言，能够实现自动化的处理。因此本文提出在 OWL-S 中改进服务描述的方式。

在物联网服务中，意图表示用户期望实现的最终目标或行动以及服务所实现的目标或行动[5]。用户通常以意图的形式表达需求，而服务提供者提供的服务描述仅表示为服务的功能描述，因此服务发现过程中存在着服务描述与用户需求之间不匹配的问题。本文通过在 OWL-S 中引入意图服务本体，以意图的方式描述服务功能，因此可以用意图的形式同时表达用户需求与服务功能，解决两者之间不匹配的问题。意图服务本体将服务意图和服务的其他信息以本体的形式结合在一起，其中意图与服务之间被定义为满足与被满足的关系，即意图满足服务。

随着物联网技术的发展，服务发现时如果仅仅考虑服务的功能(包括意图)信息，服务发现查全率和查准率会降低。当两个用户请求相同功能的服务时，不同用户对服务的上下文和 Qos 需求不同，导致服务发现时不能匹配到真正满足用户需求的服务。本文在意图服务本体中扩展物联网服务的非功能属性，包括服务的上下文与 Qos，同时服务上下文与设备上下文相关联，保证服务发现的准确率，该领域相关工作介绍见下文。

面向服务体系结构(Service Oriented Architecture, SOA)[6]现被广泛应用于服务之间的通信、组合与使用。同时，SOA 也被应用在物联网中，如文献[7]提出了一种支持互操作性、异构性、灵活性、可管理性、可扩展性和可伸缩性的基于物联网的大规模 SOA 本体；文献[8]将 SOA 和物联网的概念应用于智能家居灯，对智能设备进行控制。文献[9]在传统的 SOA 模型中引入意图，即 ISOA(Intentional Service Oriented

收稿日期：2022-02-07；**修回日期：**2022-03-30　**基金项目：**江苏省重点研发项目(BE2020084-4)

作者简介：刘星宇(1999-)，女，安徽安庆人，硕士研究生，主要研究方向为下一代通信网络技术和物联网技术；江凌云(1971-)，女(通信作者)，安徽安庆人，副教授，硕导，硕士，主要研究方向为下一代网络技术与物联网应用(jiangly@njupt.edu.cn).

Architecture)模型，介绍了基于意图的面向服务的体系结构。ISOA 为基于意图的服务描述与服务发现提供了基础的框架。

文献[10]研究了基于意图技术的最新进展，提出在自然语言理解的快速发展下，基于意图的技术有望在未来几年不断提高其适应能力。文献[11]通过在 SAWSDL(Semantic annotations for WSDL and XML schema)[12]服务描述文件中添加意图服务描述符，实现以意图的方式扩展服务描述，同时介绍了发布和匹配意图服务的框架。文献[13]从服务的文本描述中提取有关服务功能(即服务目标)的领域知识，帮助服务请求者通过意图信息查询并准确地获得相关服务，该文章使用基于自然语言解析的方法从服务的文本描述中提取服务目标，并通过度量它们的语义相似性来对服务目标进行聚类，为用户推荐服务。文献[14]介绍了一个用于存储用户意图信息的意图本体模型，提出了基于意图服务组合的体系结构。上述文献在服务描述与服务发现过程中引入了意图，但当服务种类增多时，具有相似意图和功能的服务增多，仅基于意图进行服务匹配会导致服务发现结果准确率降低。

文献[15]提出了一种基于意图的饮水服务机器人主动服务的方法，该文献在考虑用户意图的同时考虑用户的环境因素以及用户自身的上下文信息。文献[16]提出以用户需求为服务发现的中心，介绍了一种基于 OWL-S 扩展的上下文感知的有意服务发现机制。文献[17]提出了一种基于用户意图和上下文的 web 服务动态组合方法，提出基于人工智能规划技术与语义描述的 web 服务组合方式。文献[18]提出了基于上下文和用户意图的服务组合方法，在服务上下文中结合用户意图信息完成服务组合，正确响应用户需求。上述文献在服务发现过程中同时考虑服务的意图与上下文信息，但未考虑到不同用户对于服务 Qos 的需求有所不同。在服务发现过程中，缺少 Qos 的匹配会导致无法找到满足用户需求的服务。

文献[19]提出一种高效多目标自动服务组合方法，提出的模型能够将简单的原子服务组合成具有合适 Qos 的聚合服务，同时在计算成本和 Qos 之间实现更好的平衡。文献[20]对意图驱动管理系统的核心方面进行全面讨论，介绍意图的定义以及相关建模，同时考虑到服务的 Qos 性能生成端到端的系统架构。文献[21]提出了一种以需求为中心的方法,使用从 ISM(Intentional Service Model)提供的规范中提取的关键字发现服务，并自动选择相关的高 QoS 服务。上述文献在服务发现过程中同时考虑到意图以及 Qos，但未考虑到服务上下文对服务发现结果的影响。但随着服务上下文的变化以及用户上下文的变化，仅考虑意图与服务 Qos 是不够全面的，上下文同样是保证服务发现准确率的一个重要指标。

文献[22]以用户需求的结构作为输入，该结构由意图图具体化,并通过丰富该图来显示上下文信息,在服务组合时，该文献基于语义匹配技术并考虑用户上下文和 Qos 进行服务选择。文献[23]提出由用户的意图、上下文和 Qos 信息构建复合服务，同时介绍服务的描述和发布方式，匹配最合适的服务。上述文献同时考虑服务的功能和非功能参数，但未介绍非功能属性的模型和相应的服务发现机制。

以上文献大多为在互联网领域引入意图、上下文或 Qos 信息，而将其应用在物联网领域需要考虑到物联网设备的上下文信息。同时上述文献未介绍完整的引入意图、上下文和 Qos 后的服务发现机制。本文的主要贡献在于：1)提出了基于意图的服务发现框架，介绍服务发现的完整流程与服务匹配算法；2)提出在意图服务模型中扩展上下文类与 Qos 类，并在物联网服务的上下文信息中结合物联网设备及其上下文信息。

1　基于意图的服务发现框架

文章提出的服务发现框架如图 1 所示，涉及服务描述、服务注册、服务匹配以及语言解析四个实体。服务描述用于存储服务描述文件，其中本文采用 OWL-S 服务描述文件；服务注册将服务提供者提供的服务信息注册在 UDDI 服务注册库中，其中 UDDI 中存储所有服务提供商提供的服务信息；服务匹配在服务注册库中匹配符合用户需求的服务；语言解析将用户需求描述解析成意图、上下文以及 Qos 的形式，本文采用的语言解析方法为 Stanford 自然语言解析方式。在整个服务发现过程中，首先由服务提供者提供 OWL-S 服务描述文件，OWL-S 文件信息在 UDDI 服务注册库中注册并存储。同时，在用户提供需求后，由 Stanford 语言解析工具将用户需求解析成意图、上下文以及 Qos 信息，传入服务匹配中。在服务匹配中，首先通过意图匹配(基于功能属性匹配)的方式，在服务注册库中匹配满足用户意图的服务，输出由相似度排序后的服务列表再进行服务的非功能属性的匹配，包括上下文匹配与 Qos 匹配。在服务匹配工作过程中，由于服务上下文与 Qos 是实时计算的，本文需要同时更新服务在 UDDI 数据库中的信息。服务匹配完成后返回满足用户需求的服务描述文件。用户获得服务描述文件后，与服务的服务提供者进行绑定，通过服务提供者调用服务。本文提出在服务描述的 OWL-S 文件中添加服务的意图、上下文和 Qos 信息的服务描述方法，同时详细介绍服务匹配过程。

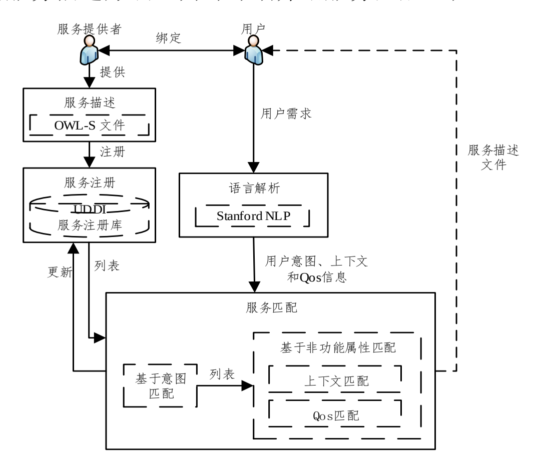


图 1　服务发现框架

Fig. 1　Service discovery framework

2　基于意图的服务描述

基于意图的服务描述主要体现在服务发现框架的服务描述部分，通过在 OWL-S 中引入意图服务本体，将服务的意图、上下文以及 Qos 信息保存在服务描述中。

2.1　结合上下文和 Qos 的意图服务本体

意图服务模型(ISM,Intentional Service Model)[24]最初由 R.S.Kaabi 提出，目的是以本体的方式存储服务信息，其中以意图的方式表示服务功能。本文在 ISM 的基础上进行上下文与 Qos 扩展，提出 CQISM(Context and Qos Intentional Service Model)本体，如图 2 所示。选择以本体表示意图服务的原因在于用本体扩展服务描述不需要更改服务描述文件的其他信息与解析工具且能规范物联网领域意图服务描述的标准。CQISM 本体模型的设计主要考虑三点：意图服务模型是对服务信息的标准化应包括服务的基本信息(意图、输入输出参数等)；由于本文的研究领域为物联网，本文添加服务上下文和服务 Qos，存储服务可用位置、状态、关联设备等信息；为了计算聚合服务的上下文与 Qos，CQISM 本体包含了服务组合类，用于保存服务类型与服务表示式。结合以上设计考虑，本文将 CQISM 本体模型分为五个类，分别为

a) 意图类：意图类表示由服务描述解析出的服务意图信息，服务意图由动词、目标、可选参数三部分组成。

b) 服务输入输出类：由服务调用前后的类以及类的状态组成，包括服务的初始情况、最终情况、前置条件和后置条件。

c) 服务组合类：表明服务的类型，包括原子服务和聚合服务，其中聚合服务由原子服务或其他聚合服务组合而成。

d) 服务上下文类：服务的上下文信息，包括服务的时间、状态、位置以及与服务相关联的设备等。

e) 服务 Qos 类：服务的 Qos 信息，包括服务的可用性、安全性、可伸缩性等。

在物联网领域中，CQISM 本体在存储服务意图的同时能够考虑到服务的非功能属性，在服务匹配过程中，通过意图、上下文与 Qos 匹配获得满足用户需求的服务，提高物联网服务发现的准确率。例如在无人机巡检系统中，无人机能够提供传输视频和图像到系统的信息管理平台、控制巡航路径等服务，而无人机所处的环境，自身的状态是不断变化的，所以本文在服务发现过程中不仅需要考虑服务的功能，还需要考虑到非功能属性，包括服务上下文、服务相关设备上下文以及服务 Qos。

下文将对 CQISM 中的不同类详细介绍。

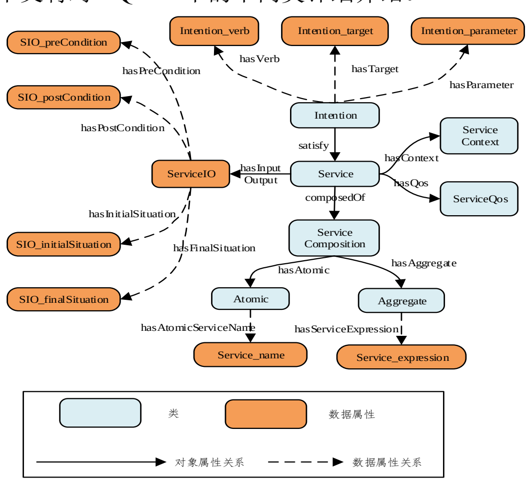


图 2　CQISM 本体模型

Fig. 2　CQISM ontology model

2.1.1 意图类

意图由能够最大程度表达服务功能或用户需求的几个词语或词组组成，意图模型如图 3 所示，该模型最初由 N.Prat 在文献[25]中提出。在意图模型中，动词表示在服务功能描述或用户需求中起主导作用的动词，目标通常是与该动词最相关的名词，可选参数由其他相关信息组成，包括方向、受益人、时间、方式、质量、数量、位置等。

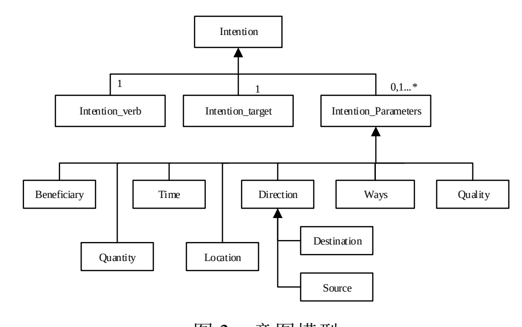


图 3　意图模型

Fig. 3　Intention model

以无人机巡检系统为例，对于服务名为 $S_{Transmit\ Image\ By\ WIFI}$ 的，服务功能为通过 WIFI 传输图像信息。服务初始情况(即输入)为空，服务最终情况(即输出)为图像信息与拍摄时间。服务意图表示为动词(Transmit)、名词(Image)以及参数(By WIFI)，其中参数属于方式类别(Ways)。

对于意图模型中动词、目标以及可选参数的提取，本文通过 Stanford Core NLP 工具包的英文依存句法分析功能，获得句子中词语的词性以及各个词语之间的关系，生成意图表达。

2.1.2 服务组合类

服务组合类分为原子服务与聚合服务。原子服务为服务的最小单元，是不可再分的，在 CQISM 模型中将原子服务名以 Service_name 字段存储在 Atomic 类中；聚合服务由原子服务或其他聚合服务组成，聚合服务的表达式以 Service_expression 字段存储在 Aggregate 类中。聚合服务表达式的生成需要结合服务的 MAP 图与服务的不同组合方式，下面分别介绍 MAP 图与服务组合方式。

1) MAP 图

MAP 图是一个有标记且有方向的图，图 4 所示为 MAP 图的最小单元，其中 Start 为源意图，Stop 为目的意图，Strategy(策略)为边。MAP 图可用来表示所有原子或聚合服务，且 MAP 图中的聚合服务可细化为其他 MAP 图表示的子聚合服务，可以理解为高级服务意图可被分解为其他低级服务意图，直到聚合服务全部被分解为原子服务为止。通过计算 MAP 图中所有路径可以确定该 MAP 图表示的聚合服务的表达式。



图 4　MAP 图最小单元

Fig. 4　MAP Minimum unit of the MAP

在物联网服务中，以无人机巡检系统为例，对于服务名为 $S_{Perform\ Inspection}$ 的聚合服务，服务功能为使无人机执行检查任务，MAP 图如图 5 所示。图 5 中 MAP 图表示要使无人机执行检查任务，首先需要通过电池电量计检查无人机剩余电量，再采集信息，采集信息的方式包括未指定位置采集和指定位置采集两种策略。

在图 5 中 $S_{Collect\ Information\ By\ non-designated\ position}$ 与 $S_{Collect\ Information\ By\ designated\ position}$ 为聚合服务。以 $S_{Collect\ Information\ By\ designated\ position}$ 为例，该服务功能为采集某一给定位置的信息，MAP 图如图 6 所示，其中 Start 表示图 5 中 Check Batterylevel，Stop 表示图 5 中 Collect Information。以图 6 所示，无人机首先需要飞行至某一给定位置，再传输位置信息与图像信息，其中传输位置信息可通过绝对位置和相对位置两种方式，传输图像信息可通过 OFDM(Orthogonal Frequency Division Multiple--xing)和 WIFI 两种方式。

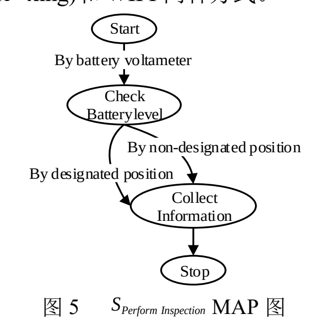


图 5　$S_{Perform\ Inspection}$ MAP 图

Fig. 5　$S_{Perform\ Inspection}$ MAP figure

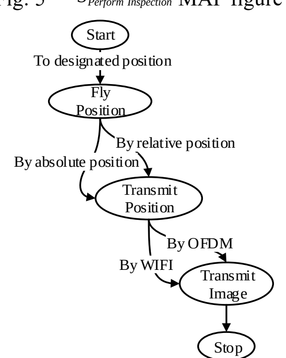


图 6　$S_{Collect\ Information\ By\ designated\ position}$ MAP 图

Fig. 6　$S_{Collect\ Information\ By\ designated\ position}$ MAP figure

2) 服务组合方式

服务与服务之间可以通过不同的操作符联系生成新的聚合服务，聚合服务包括复合服务和变体服务两类，下面分别介绍这两种聚合服务的表示方式。

复合服务指服务与服务之间的组合关系是顺序的或并行的。顺序执行的复合服务表示其分解后的服务必须按照一定的顺序执行，使用操作符“•”表示，式(1)表示图 6 中顺序执行的复合服务；并行执行的复合服务表示分解后的服务不需要考虑执行顺序，可同时执行直至完成，使用操作符“//”表示，式(2)表示图 6 中服务名为 $S_{Transmit\ Image\ By\ WIFI}$ 的服务分解后的并行表示。

$$S_{Collect\ Information\ By\ designated\ position} = \bullet(S_{Fly\ Position\ To\ designated\ position}, S_{Transmit\ Position}, S_{Transmit\ Image}) \tag{1}$$

$$S_{Transmit\ Image\ By\ WIFI} = //(S_{Transmit\ Time\ By\ WIFI}, S_{Transmit\ Picture\ By\ WIFI}) \tag{2}$$

变体服务包括三种类型，分别为选择变体服务、替代变体服务、多径变体服务。选择变体服务表明服务可通过不同的策略实现，但这些策略并不相斥，用操作符“∨”表示。例如，图 6 中传输位置信息可通过绝对位置和相对位置两种策略，但两种策略可同时选择也可选择其中一种，其表示如式(3)所示；替代变体服务指完成一个服务的不同策略之间为互斥关系，由操作符“⊗”表示。例如，图 5 中无人机执行收集信息的服务时，可通过未指定位置收集和指定位置收集两种策略，但这两种策略只能选择一种，其表式(4)所示；多径变体服务指对于一个变体服务，可通过多个不同路径实现相同的意图，由操作符“∪”表示，对于简单 MAP 图表示的聚合服务，采用多径变体服务表示方式更加便于服务发现时上下文和 Qos 的计算。

$$S_{Transmit\ Position} = \vee(S_{Transmit\ Position\ By\ absolute\ position}, S_{Transmit\ Position\ By\ relative\ position}) \tag{3}$$

$$S_{Collect\ Information} = \otimes(S_{Collect\ Information\ By\ designated\ position}, S_{Collect\ Information\ By\ non\text{-}designated\ position}) \tag{4}$$

3) 聚合服务表达式的生成

OWL-S 文件中的 Process Model 部分包含服务的所有进程信息，可以从中提取 MAP 图的信息，通过 MAP 图可以完整地表示一个复杂的聚合服务，再通过 MAP 图中的所有路径生成聚合服务的表达式。由于本文的研究领域是物联网，考虑到物联网服务大多为聚合服务，而服务的上下文与 Qos 的计算需要考虑到该聚合服务的子服务以及与子服务相关联的物联网设备，所以本文通过聚合服务的表达式计算上下文与 Qos。上文图 5 与图 6 中 MAP 图的表达式如表 1 与表 2 所示(符号“*”表示服务被重复调用)，表格中原子服务指 MAP 图所涉及到的所有原子服务名，聚合服务指 MAP 图所涉及到的所有聚合服务的表达式。

2.1.3 上下文与 Qos 类

在服务发现过程中，传统的基于 OWL-S 语言的语义化描述以及本文基于意图服务本体的服务描述，都需要根据服务功能进行匹配，寻求最符合用户需求的服务。为提高物联网服务发现的准确率，本文同时考虑服务的非功能属性，包括服务的上下文类与 Qos 类。

1) 上下文类

物联网服务与传统 web 服务不同在于，物联网服务涉及各种设备，包括传感器、执行器等，物联网服务与其相关设备联系紧密，因此本文在物联网服务上下文类中添加 SC_attachedDevice 字段，见图 7。原子服务上下文信息与单个设备上下文信息由服务提供商提供，聚合服务上下文信息包含子服务的上下文信息以及与子服务相关联的设备。

表 1　$S_{Perform\ Inspection}$ 服务表达式

Tab. 1　$S_{Perform\ Inspection}$ Service expression

原子服务	$S_{Check\ Batterylevel\ By\ battery\ voltameter}$, $S_{Transmit\ Position\ By\ path}$, $S_{Transmit\ Image\ By\ Path}$
聚合服务	$S_{Perform\ Inspection} = \cup(\bullet(S_{Check\ Batterylevel\ By\ battery\ voltameter}$, $S_{Collect\ Information\ By\ designated\ position}), \bullet$ $(S_{Check\ Batterylevel\ By\ battery\ voltameter}$, $S_{Collect\ Information\ By\ non\text{-}designated\ position}))$ $S_{Collect\ Information\ By\ non\text{-}designated\ position} =$ $//(S_{Transmit\ Position\ By\ path}$, $S_{Transmit\ Image\ By\ Path})^{*}$

表 2　$S_{Collect\ Information\ By\ designated\ position}$ 服务表达式

Tab. 2　$S_{Collect\ Information\ By\ designated\ position}$ service expression

原子服务	$S_{Fly\ Position\ To\ designated\ position}$, $S_{Transmit\ Position\ By\ absolute\ position}$, $S_{Transmit\ Position\ By\ relative\ position}$, $S_{Transmit\ Time\ By\ WIFI}$, $S_{Transmit\ Picture\ By\ WIFI}$, $S_{Transmit\ Time\ By\ OFDM}$, $S_{Transmit\ Picture\ By\ OFDM}$
聚合服务	$S_{Collect\ Information\ By\ designated\ position} =$ $\bullet(S_{Fly\ Position\ To\ designated\ position}, S_{Transmit\ Position}, S_{Transmit\ Image})$ $S_{Transmit\ Position} =$ $\vee(S_{Transmit\ Position\ By\ absolute\ position}, S_{Transmit\ Position\ By\ relative\ position})$ $S_{Transmit\ Image} =$ $\otimes(S_{Transmit\ Image\ By\ WIFI}, S_{Transmit\ Image\ By\ OFDM})$ $S_{Transmit\ Image\ By\ WIFI} =$ $//(S_{Transmit\ Time\ By\ WIFI}, S_{Transmit\ Picture\ By\ WIFI})$ $S_{Transmit\ Image\ By\ OFDM} =$ $//(S_{Transmit\ Time\ By\ OFDM}, S_{Transmit\ Picture\ By\ OFDM})$

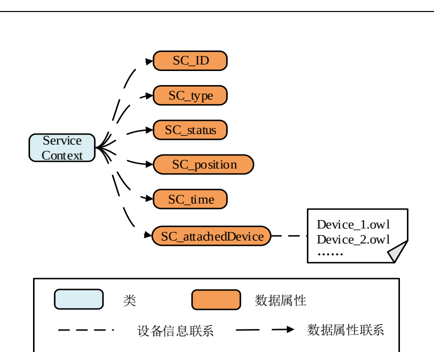


图 7　ServiceContext 类

Fig. 7　Servicecontext class

图 7 为物联网服务上下文类，包括服务标识符(SC_ID，服务唯一标识)、服务类型(SC_type，服务是原子服务还是聚合服务)、服务状态(SC_status，服务可用性)、服务时间(SC_time，服务能被调用的时间段)、服务位置(SC_position，服务能被调用的位置)、服务附加设备(SC_attached Device，与服务相关联的设备)。在服务通过上下文匹配时，需要实时的利用设备上下文更新服务上下文信息，其中设备上下文信息存储在设备本体(OWL 文件)中。设备本体模型见图 8，包括设备属性与设备能力，其中设备属性包括静态属性与动态属性，静态属性存储设备固定信息，动态属性包括设备的能源、状态、位置和时间等。设备能力包括通信能力、控制能力、正在发生的事件和展示能力。在服务上下文计算过程中，设备的动态属性与服务的上下文相关，例如设备状态可用以及剩余能源足够时，服务上下文状态(SC_Status)才能为可用。

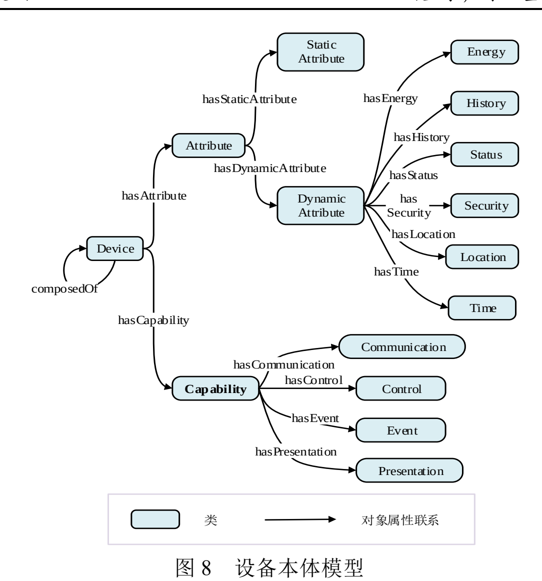


图 8　设备本体模型

Fig. 8　Device ontology model

2) Qos 类

考虑到用户对于服务质量的需求不同(例如，用户请求的服务需要很高的安全性，而对服务的其他 Qos 属性要求不高)，本文在 CQISM 中扩展 ServiceQos 类存储服务质量信息。在用户请求服务时，通过意图与上下文匹配到满足需求的服务后，再通过服务质量匹配，选择满足 Qos 需求的服务返回给用户。物联网服务 Qos 类及关联指标如图 9 所示(图 9 中模型参考文献[26]中物联网服务 Qos 指标的相关内容)。图 9 中 Qos 类包括物联网服务的可用性、性能、安全性、可伸缩性、可靠性、互操作性以及可用性。对于 Qos 的评价指标，本文采用定性评价方式，分别为数字“1”、“2”、“3”、“4”，分别表示“很低”、“低”、“高”、“很高”。通过四个层级的方式表示 Qos 的目的是统一服务提供者与用户对 Qos 的表示方式。Qos 的各个指标的计算方式不同，每个 Qos 指标的层级度量方式由服务提供者制定。原子服务 Qos 信息由服务提供商直接提供，聚合服务的 Qos 需要考虑到组成该聚合服务的子服务的 Qos 信息。

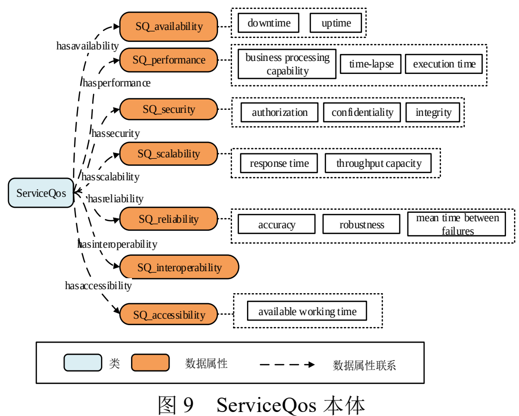


图 9　ServiceQos 本体

Fig. 9　Serviceqos ontology

2.2　OWL-S 服务描述文件的扩展

OWL-S 服务描述文件分为三个主要部分：服务配置文件(Service profile)，用于描述服务做什么,包括服务的名称、描述、服务质量以及服务发布商等信息；进程模型(Process model)，用于描述如何使用该服务以及服务调用后的结果；服务基础(Grounding)，用于描述如何访问服务，包括通信协议、消息格式以及端口号等[27]。

服务描述扩展指的是在基础语义化服务描述中添加本文提出的 CQISM 本体信息生成扩展后的服务描述文件，扩展框架图如图 10 所示。由图 10 可知，本文在 OWL-S 服务描述文件的 ServiceProfile 部分扩展 CQISM 本体信息。在 OWL-S 中，原子服务与聚合服务的信息均存储在 ProcessModel 中，本文从中提取原子服务与聚合服务的组成关系，生成 MAP 图，再由 MAP 图生成服务组合表达式，将表达式以 ServiceComposition 类的形式存储在 CQISM 本体中。扩展后的服务描述文件能够丰富 OWL-S 服务描述语言的语义，解决服务提供者提供的服务描述与用户需求描述之间概念不匹配的问题，同时考虑服务的非功能属性，提高服务发现的准确率。

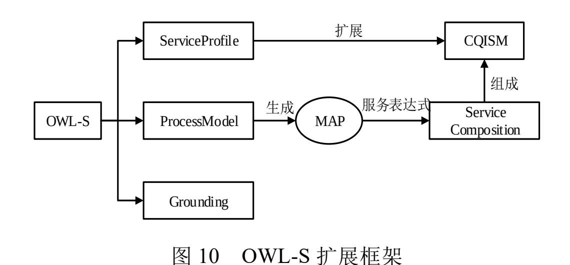


图 10　OWL-S 扩展框架

Fig. 10　OWL-S extension framework

3　基于意图的服务匹配

基于意图的服务匹配在图 1 的服务发现框架中作用在服务匹配部分，该部分实现从 UDDI 服务注册库中通过功能与非功能属性匹配返回满足用户需求的服务。

在服务匹配工作之前，需要在 OWL-S 文件中提取 CQISM 本体，通过解析 CQISM 本体，提取服务的意图、上下文以及 Qos 信息并存储在服务注册库中。在用户输入需求时，本文首先需要通过 stanford NLP 解析用户的意图，进行意图匹配，见算法 1。意图匹配完成后，根据相似度结果进行排序，选择符合用户意图需求的服务生成新的表格，再进行上下文匹配和 Qos 匹配。

由于服务包括原子服务与聚合服务，并且涉及到各种物联网设备，所以需要对完成意图匹配后的服务进行上下文计算和匹配，见算法 2。如果服务为原子服务，服务上下文与设备上下文相关联，算法 2 中第 4 行设备上下文计算遵循以下原则：a) 设备的状态、位置、时间等字符型表示属性采用交集∩的方式计算，即当一个原子服务有 j 个相关联设备(Device_1、Device_2 ……Device_j)时，Device_j 某一属性 q 的值为 vj 表示为$<q,v_j>_{Device_j}$，则该原子服务的 q 属性的值为$v=v_1\cap v_2\cap\ldots\cap v_j$，表示为$<q,v>_{Atomic}$；b) 设备的能量和安全性等数字型表示属性采用取最小值的方式计算即$v=\min(v_1,v_2\ldots,v_j)$。例如一个原子服务关联的两个设备的位置信息分别表示为$<Position,NJ、SH>_{Device_1}$，$<Position,NJ>_{Device_2}$，则该原子服务的位置信息表示为$<Position,NJ>_{Atomic}$。如果服务为聚合服务，服务上下文与其子服务上下文有关，其子服务之间的关系由聚合服务表达式表示，算法 2 中第 9 行服务上下文的计算遵循以下规则：

如果是复合服务，即聚合服务的表达公式中有 • 或 / / ，采用交集∩与最小值的计算方式计算服务上下文。

如果是变体服务，即聚合服务的公式中有 ⊗ 、∨ 或∪，采用并集∪与最大值的方式计算上下文。当聚合服务的表达式中涉及其他聚合服务，仍采用规则 a)与 b)，直到聚合服务中所有子服务都被分解为原子服务即完成聚合服务上下文的计算。

服务上下文实时的计算并不涉及系统中的所有服务，而是在基于意图进行排序后，选取与用户需求相似度较高的服务进行计算，这样做的目的是在保证准确率的情况下，避免服务发现的时间过长。完成上下文计算后，本文通过算法 2 伪代码中第 14 至 20 行，选择符合用户上下文需求的服务。

同样，在进行 Qos 匹配前，需要对已完成意图和上下文匹配后的服务进行 Qos 计算。原子服务的 Qos 由服务提供商提供，聚合服务的 Qos 计算与上下文计算类似，本文不再详

细展开(注意：Qos 计算不考虑服务关联设备)。

算法 1 意图匹配算法

输入：含意图信息服务列表(List_Intention)，用户功能需求
输出：基于意图匹配后的服务列表
/*Sj 表示列表中的每个服务*/
1.从用户功能需求中提取意图信息分别为：user_verb,user_target,user_parameter
2.For each Sj in List_Intention do
3.从 Sj 中提取该服务的动词、目标和可选参数，分别为：verb、target、parameter
4.分别计算 verb 与 user_verb、target 与 user_target、parameter 与 user_parameter 语义相似度，并求和 sum
5.End For
6.以 sum 值由大到小对 List_Intention 列表排序
7.从 List_Intention 列表中选取前 100 个服务生成新列表 List_ServiceContextSelected
8.返回基于意图匹配后的服务列表(List_ServiceContextSelected)

算法 2 上下文计算与匹配算法

输入：用户上下文需求(userContext)，包含上下文信息的服务列表(List_ServiceContext)，基于意图匹配后的服务列表(List_ServiceContextSelected)
输出：基于意图与上下文匹配后的服务列表(List_ServiceContextQosSelected)
/*Sj 表示列表中的每个服务，SC 表示某一个服务的每类上下文信息，UC 表示用户上下文需求的每类上下文信息*/
1.For each Sj in List_ServiceContext Selected do
2.if Type(Sj)==’Atomic’:
3.从 Sj 中提取该服务关联设备
4.计算服务关联设备上下文
5.利用设备上下文信息更新 Sj 在服务列表 List_ServiceContext Selected 中的信息
6. End if
7.if Type(Sj)==’Aggregate’:
8.从 Sj 中提取聚合服务表达式 Service_e xpression
9.由聚合服务表达式计算该聚合服务上下文
10.利用计算后的上下文信息更新 Sj 在服务列表 List_ServiceContextSelected 中的信息
11.End if
12.End For
13.利用 List_ServiceContextSelected 中实时计算的上下文信息更新包含上下文信息的服务列表 List_ServiceContext
14.For each Sj in List_ServiceContext Selected do
15.For each SC in Sj do
16.For each UC in userContext do
17. if Name(UC)=Name(SC) and Value(UC)⊆Value(SC)
/*假设 Name(UCj)=Name(SCj)=“Position”*/
18.将服务信息存储在满足位置条件的服务列表 list_Position 中
19.结束所有 if 与 for 循环
20.List_ServiceContextQosSelected 结果为 List_Positionn List_StatusnList _TimenList_Energy
21.返回完成上下文计算与匹配后的服务列表 List_Service ContextQosSelected

4 仿真与结果分析

4.1 开发环境

本文开发语言为 python 语言，python 语言作为可跨平台的语言，可以运行在 Windows、Mac 和各种 Linux/Unix 系统上；python 语言开发工具为 anaconda 下的 jupyter notebook；本体构建工具为 Protégé-5.5.0，用于构建 CQISM 本体；语言解析工具为 Stanford NLP，用于提取意图信息；测试数据集为 OWLS-TC_v2.2_revision_3，该数据集为服务描述文件的集合，包括医疗、教育、食品、工业等领域。

4.2 服务描述文件的扩展

以无人机通过 WIFI 传输图像的物联网服务为例，表 3 为在 OWL-S 文件中扩展 CQISM 本体后的部分服务描述文件内容，其中<profile:CQISMPath>中为 CQISM 本体的路径，<profile:CQISMIRI>中为本体的 IRI(Internationalized Resource Identifiers，国际化资源标识符)，本体的路径与 IRI 均可唯一表示一个意图服务本体，可利用 python 代码解析 OWL-S 文件，提取相应 CQISM 本体并解析。

通过解析 CQISM 本体能够获得服务的意图、上下文、聚合服务表示式以及 Qos 信息，这些信息将用于下一步的服务匹配过程。CQISM 本体部分信息见表 4，其中第 3、4、5 行表示服务意图信息(动词、目标与参数)；第 9、10 行表示服务上下文信息，包括可用状态(1 表示服务当前可用)以及相关设备；第 14 行表示服务 Qos 信息(表中表示可靠性层级为 4)。

考虑到物联网服务中涉及各种设备，在服务上下文匹配时需要考虑到相关设备上下文的影响，因此系统中需要存储设备本体文件用于更新服务上下文。以摄像头设备(Camera_1)为例，其本体文件部分信息如表 5 所示，表 5 展示了设备上下文中的能源信息，由表中第 7 行可知，该设备剩余能源为 80%。

表 3 TransmitPictureByWIFI_service.owls 文件

Tab. 3 Transmitpicturebywifi_service.owls file

```
<profile:Profile rdf:ID="TransmitPictureByWIFI_Profile">
<profile:CQISMIRI>
http://www.semanticweb.org/administrator/ontologies/2021/10/untitled-ontology-73
</profile:CQISMIRI>
<profile:CQISMPath>
E:\code\Intention\Intention_50\ISM_50\TransmitPictureByWIFI_servic
e.owl
</profile:CQISMPath>
</profile:Profile>
```

表 4 TransmitPictureByWIFI_service.owl(CQISM 本体)

Tab. 4 Transmitpicturebywifi_service.owl(CQISM ontology)

```
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/administra tor/
ontologies/2021/10/untitled-ontology-73#Intention">
<Intention_parameter>by WIFI,at certain position
</Intention_parameter>
<Intention_target>picture</Intention_target>
<Intention_verb>transmit</Intention_verb>
</owl:NamedIndividual>
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/administra tor/
ontologies/2021/10/untitled-ontology-73#ServiceContext">
<SC_status>1</SC_status>
<SC_attachedDevice>Camera_1</SC_attachedDevice >
</owl:NamedIndividual>
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/administra tor/
ontologies/2021/10/untitled-ontology-73#ServiceQos">
<SQ_reliability rdf:datatype="http://www.w3.org/2001/XMLSchema
#integer">4</SQ_reliability>
</owl:NamedIndividual>
```

表 5　Camera_1.owl(设备本体)

Tab. 5　Camera_1.owl(device ontology)

```
<owl:NamedIndividual
        rdf:about="http://www.semanticweb.org/administra tor/
                        ontologies/2021/10/untitled-ontology-59#Energy">
<rdf:type rdf:resource="http://www.semanticweb.org/administrator/
                    ontologies/ 2021/10/untitled-ontology-59#Energy"/>
<Device_id>IOT/Entity/UAV/Camera/Camera_0001</Device_id>
<Energy_value rdf:datatype="http://www.w3.org/2001/XML
Schema#float"> 0.8</Energy_value >
</owl:NamedIndividual>
```

4.3　基于意图的服务发现仿真结果分析

4.3.1 服务发现准确率

本文首先验证仅考虑意图与服务功能对服务发现准确率的影响，对于不同的系统服务数量使用相同服务请求输入，这些服务请求仅包括服务的功能信息，用式(5)计算准确率，实验结果见图 11。实验结果表明，在相同服务请求不同服务数量的情况下，基于意图描述方式的准确率一直高于基于服务功能描述方式的准确率，且基于服务功能描述的准确率会随着系统总服务个数的增加，下降的趋势较大。本文取系统中服务数量分别为 200,400,800 时的准确率取均值进行比较，基于服务功能描述的服务发现与基于意图的服务发现准确率分别为 77.5%，88.3%，表明在不考虑其他影响服务发现准确率因素的情况下，基于意图的方式能提高 10.8%的准确率，可得用意图提取的方式表示服务描述是完全可行的且本文提出的方法能很好的提取用户的意图。

$$准确率=\frac{请求结果正确次数}{服务请求总次数} \tag{5}$$

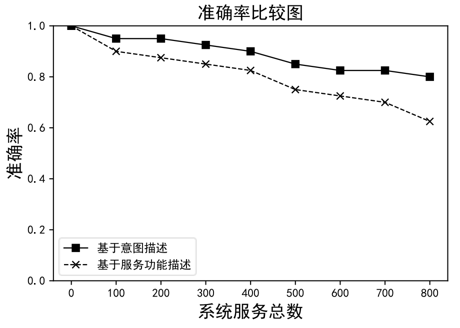


图 11　服务意图描述与服务功能描述准确率比较

Fig. 11　Service intention description and service function description accuracy comparison

在物联网服务发现过程中，仅考虑服务意图描述或服务功能是不全面的。传统的服务匹配方式是在服务功能描述的基础上结合输入和输出信息，如 OWLS-MX[28]服务匹配器。基于意图服务匹配方式是在服务意图的基础上结合上下文和 Qos 信息，例如，文献[16]在意图的基础上结合上下文，文献[21]在意图的基础上结合 Qos。图 12 将 OWLS-MX 匹配器、文献[16](意图+上下文)、文献[21](意图+Qos)以及本文(意图+上下文+Qos)三种服务匹配方式准确率比较，式(5)为准确率计算公式。取系统中服务数量分别为 200,400,800 时的准确率取均值进行比较，意图+上下文+Qos、OWLS-MX、意图+上下文、意图+Qos 准确率均值分别为 84%、77.3%、80%、73%。表明在服务发现过程中同时考虑服务功能与非功能属性能提高服务发现的准确率。

4.3.2 服务发现时间

表 6 为在系统数量分别为 200、400 和 800 时基于意图的服务发现时间与 OWLS-MX 匹配器服务发现时间的具体数值比较(表中仅展示 10 次服务请求的时间)，其中服务发现时间单位为秒，R_1 表示第一次服务请求，依此类推。通过式(6)计算不同系统数量情况下的服务发现平均时间，其中 T_j 表示第 j 次请求查询时间，时间比较图如图 13 所示，随着服务数量的增加两者的时间均在增加，但基于意图的服务发现所需要的查询时间更长。这是由于随着服务数量增加，CQISM 本体数量相应增多，意图匹配相对于功能匹配所涉及到的参数更多，且意图匹配多出一步在用户需求中提取意图的步骤。取系统中服务数量分别为 200,400,800 时的时间取均值进行比较，OWLS-MX 消耗时间与基于意图方式消耗时间均值分别为 7.0951s，8.8865s，可得基于意图的服务发现平均多消耗 1.7914s 的时间。

$$服务发现平均时间=\frac{\sum_{j=1}^{服务请求总次数} T_j}{服务请求总次数} \tag{6}$$

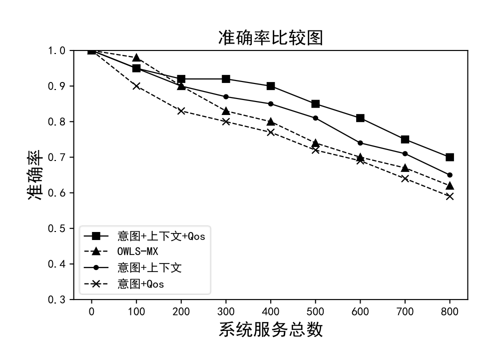


图 12　服务发现准确率比较

Fig. 12　Service discovery accuracy comparison

表 6　服务发现时间比较(单位：秒)

Fig.6　service discovery time comparison(Unit: second)

服务请求	意图 (200 服务)	OWLS-MX (200 服务)	意图 (400 服务)	OWLS-MX (400 服务)	意图 (800 服务)	OWLS-MX (800 服务)
R_1	3.3437	2.104	7.4843	6.1444	15.3343	13.6608
R_2	3.4531	2.1152	7.7031	5.2345	17.822	11.3251
R_3	3.3987	3.0716	6.2367	6.1869	14.7888	12.9813
R_4	3.425	3.0577	8.4062	7.0156	18.7552	14.772
R_5	3.4531	3.0377	7.0775	6.5516	15.7223	13.8627
R_6	3.375	2.0941	7.5156	4.8329	16.0347	11.2862
R_7	3.2531	2.4616	7.0863	6.7978	15.6558	14.2356
R_8	3.3125	2.0695	8.0562	5.0253	17.2954	12.8343
R_9	3.3593	2.3140	8.7187	6.6189	18.6226	15.4418
R_10	3.2375	2.4577	9.1406	6.1636	19.0431	13.8173

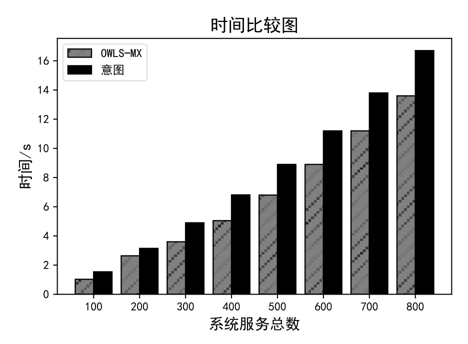


图 13　服务发现时间比较

Fig. 13　Service discovery time comparison

在基于意图的服务匹配后，生成意图相似度排序表格，本文选取意图相似度高的一些服务再次进行上下文以及 Qos 的匹配。以选取相似度高的前 100 个服务为例，在提出 20 次不同请求时相应的上下文查询时间和 Qos 查询时间如表 7 所示。取 20 次时间的均值，上下文和 Qos 时间的平均值分别为 0.268015s，0.06888s。可以认为这些时间对于图 13 中意图

的查询时间来说影响很小，但通过上下文与 Qos 匹配能提高服务发现的准确率。

表 7 上下文与 Qos 服务发现时间

Tab. 7 Context and Qos service discovery time

服务请求	上下文时间(s)	Qos时间(s)	服务请求	上下文时间(s)	Qos时间(s)
R_1	0.2612	0.0439	R_11	0.2314	0.1516
R_2	0.1421	0.0418	R_12	0.47	0.158
R_3	0.2295	0.1307	R_13	0.3362	0.0359
R_4	0.3316	0.0424	R_14	0.2328	0.0558
R_5	0.1317	0.0419	R_15	0.2757	0.0368
R_6	0.272	0.0379	R_16	0.2296	0.0294
R_7	0.271	0.0431	R_17	0.2401	0.0369
R_8	0.3321	0.0708	R_18	0.3639	0.1537
R_9	0.1766	0.04	R_19	0.227	0.1443
R_10	0.271	0.0425	R_20	0.3348	0.0402

5 结束语

本文提出一种基于意图、上下文与 Qos 的服务描述方法以及服务匹配算法，以意图服务本体(CQISM)的方式存储服务意图、上下文以及 Qos 信息，并将本体保存在 OWL-S 服务描述文件中用于服务匹配。意图的提取能够解决服务描述与用户需求描述之间不匹配的问题；上下文和 Qos 信息的引入能够进一步提高服务发现的准确率；本文采用本体作为工具描述意图服务：将 CQISM 本体引入服务描述能够规范意图服务本体的描述标准，同时扩展服务描述文件时不需要更改原本的服务描述信息，同时不需要更换用于解析 OWL-S 文件的工具。文章仿真部分的对比文献[16]与[21]均采用本体的方式描述意图服务，实验结果表明本文在意图基础上同时考虑上下文与 Qos 能够在较短时间内提高服务发现的准确率。未来考虑改进服务匹配算法，规范服务发现机制，进一步缩短基于意图的服务发现时间，同时改进基于意图的服务发现在物联网领域中的应用。

参考文献：

[1] Chinnici R, Moreau J J, Ryman A, *et al.* Web Services Description Language (WSDL) version 2. 0 part 1: core language [EB/OL]. (2007) [2022-01-15]. http://www. w3. org/TR/2007/REC-wsdl20-20070626.

[2] Martin D, Burstein M, Hobbs J, *et al.* OWL-S: semantic markup for web services [J/OL]. W3C member submission, 2004, 22 (4) . (2004-11-01) [2022-01-15]. http://www. daml. org/services/owl-s/1. 1/overview/.

[3] De Bruijn J, Bussler C, Domingue J, *et al.* Web Service Modeling Ontology (WSMO) [J]. Interface, 2005, 5 (1): 50.

[4] Akkiraju R, Farell J, Miller J A, *et al.* Web service semantics-WSDL-S [EB/OL]. (2005-11-07) [2022-01-15]. https://www. w3. org/Submission/2005/SUBM-WSDL-S-20051107/.

[5] Ponce V, Abdulrazak B. Intention as a context: an activity intention model for adaptable development of applications in the Internet of Things [J]. IEEE Access, 2021 (9): 151167-151185.

[6] Perrey R, Lycett M. Service-oriented architecture [C]// 2003 Symposium on Applications and the Internet Workshops. Piscataway, NJ: IEEE Press, 2003: 116-119.

[7] Mishra S K, Sarkar A. Service-oriented architecture for internet of things: a semantic approach [J/OL]. Journal of King Saud University-Computer and Information Sciences, 2021, 2021 (24): 1-12. (2021) [2022-01-15]. https://doi. org/10. 1016/j. jksuci. 2021. 09. 024.

[8] Irawan Y, Linarta A, Febriani A, *et al.* Smart home light based service oriented architecture and IoT [J]. Journal of Physics: Conference Series, 2021, 1845 (1): 012070.

[9] Rolland C, Kirsch-Pinheiro M, Souveyet C. An intentional approach to service engineering [J]. IEEE Transactions on Services Computing, 2010, 3 (4): 292-305.

[10] Zeydan E, Turk Y. Recent advances in intent-based networking: a survey [C]// 2020 IEEE the 91st Vehicular Technology Conference. Piscataway, NJ: IEEE Press, 2020: 1-5.

[11] Aljoumaa K, Assar S, Souveyet C. Publishing intentional services using extended semantic annotation [C]// 2011 the 5th International Conference on Research Challenges in Information Science. Piscataway, NJ: IEEE Press, 2011: 1-9.

[12] Kopecky J, Vitvar T, Bournez C, *et al.* Sawsdl: semantic annotations for wsdl and xml schema [J]. IEEE Internet Computing, 2007, 11 (6): 60-67.

[13] Zhang Neng, Wang Jian, Ma Yutao, *et al.* Web service discovery based on goal-oriented query expansion [J]. The Journal of Systems and Software, 2018, 142: 73-91.

[14] Daosabah A, Guermah H, Nassar M. PDDL planning and ontologies, a tool for automatic composition of intentional-contextual web services [M]// Ouaissa and Mariya, Computational Intelligence in Recent Communication Networks. Switzerland: Springer, Cham, 2022: 163-190.

[15] Hao Man, Cao Weihua, Wu Min, *et al.* An initiative service method based on fuzzy analytical hierarchy process and context intention inference for drinking service robot [J]. IEEE Transactions on Cognitive and Developmental Systems, 2018, 11 (2): 221-233.

[16] Najar S, Pinheiro M K, Souveyet C, *et al.* Service discovery mechanism for an intentional pervasive information system [C]// 2012 IEEE the 19th International Conference on Web Services. Piscataway, NJ: IEEE Press, 2012: 520-527.

[17] Daosabah A, Guermah H, Choukri I, *et al.* Integrating context and intention for optimal semantic web service composition using AI planning [C]// 2021 the 4th International Conference on Advanced Communication Technologies and Networking (CommNet) . Piscataway, NJ: IEEE Press, 2021: 1-9.

[18] Daosabah A, Guermah H, Nassar M. Dynamic composition of services: an overview of approaches led by the context and intent of the user [C]// Proceedings of the 4th International Conference on Big Data and Internet of Things. New York: ACM Press, 2019: 1-8.

[19] Wang Zhaoning, Cheng Bo, Zhang Wenkai, *et al.* Many-objective automatic service composition based on temporal goal decomposition [J]. IEEE Transactions on Network and Service Management, 2021, 18 (3): 3813-3828.

[20] Mwanje S S, Banerjee A, Goerge J, *et al.* Intent-driven network and service management: definitions, modeling and implementation [J/OL]. TechRxiv, 2021, 2021 (1): Article ID 17075450. v1. (2021-12-06) [2022-01-15]. https://doi. org/10. 36227/techrxiv. 17075450. v1.

[21] Driss M, Moha N, Jamoussi Y, *et al.* A requirement-centric approach to web service modeling, discovery, and selection [C]// International conference on service-oriented computing. Berlin, Heidelberg: Springer, 2010: 258-272.

[22] Fki E, Tazi S, Drira K. Automated and flexible composition based on abstract services for a better adaptation to user intentions [J]. Future Generation Computer Systems, 2017, 2017 (68): 376-390.

[23] Khanfir E, Djmeaa R B, Amous I. Automated publish, discovery and composition of intentional web services adaptable to both quality and context [C]// 2016 IEEE the 18th International Conference on High Performance Computing and Communications. Piscataway, NJ: IEEE Press, 2016: 639-646.

[24] Kaabi R S. A methodological approach for modeling and operationalzing

intentional services [D]. Paris: Universite of Paris 1 La Sorbonne, 2007.

[25] Prat N. Goal formalisation and classification for requirements engineering [C]// Requirements Engineering: Foundation for Software Quality. Paris: archive ouverte HAL, 1997: 145-156.

[26] Singh M, Baranwal G. Quality of service (qos) in internet of things [C]// 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU) . Piscataway, NJ: IEEE Press, 2018: 1-6.

[27] GAO Yixiang. Research and design of QoS simulation analysis tool for composite web services based on OWL-S [D]. Harbin: Harbin Institute of Technology, 2016.

[28] 高一翔. 基于 OWL-S 的组合 Web 服务 QoS 仿真分析工具的研究与设计 [D]. 哈尔滨: 哈尔滨工业大学, 2016.

[29] Klusch M, Fries B, Sycara K. OWLS-MX: A hybrid semantic web service matchmaker for OWL-S services [J]. Journal of Web Semantics, 2009, 7 (2): 121-133.